

Convolutional neural networks performance evaluation applied to automated pavement crack detection

Avaliação do desempenho de redes neurais convolucionais para detecção automática de trincas em pavimentos

Francisco Dalla Rosa¹, Laura Dall'Igna Favretto², Vítor Borba Rodrigues³, Nasir G. Gharaibeh⁴

¹Federal University of Passo Fundo, Rio Grande do Sul – Brazil, dallarosa@upf.br

²Federal University of Passo Fundo, Rio Grande do Sul – Brazil, 142607@upf.br

³Federal University of Passo Fundo, Rio Grande do Sul – Brazil, 111302@upf.br

⁴Texas A&M University, Texas – Estados Unidos, ngharaibeh@civil.tamu.edu

Recebido:

21 de janeiro de 2020

Aceito para publicação:

19 de maio de 2020

Publicado:

15 de dezembro de 2020

Editor de área:

Jorge Barbosa Soares

Keywords:

Convolutional neural network.
Pavement management systems.
Automated pavement crack
detection.
Computing vision.

Palavras-chaves:

Redes Neurais Convolucionais.
Gerência de Pavimentos.
Detecção automatizada de trincas
em pavimentos.
Visão computacional.

DOI:10.14295/transportes.v28i5.2283

**ABSTRACT**

This research aims to analyze the performance of Convolutional Neural Network (CNN) as an automated tool applied to pavement surface crack detection. A group of pictures from different segments of chip seal pavement, acquired from photographic recording systems mounted on specific vehicles, was evaluated. An open-source machine learning library PyTorch available in the Python script language was applied to evaluate the images. The influence of three techniques used to process the pictures and the complexity of neural networks on the crack identification performance are discussed as well. The accuracy, precision, recall, and F1 score metrics were used to assess the neural network performance. The results show a good performance of the selected algorithm for pavement crack detection based on the observed metrics. Furthermore, it was found that the complexity of the neural network is an important factor that should be considered during the image classification process.

RESUMO

Neste artigo é avaliado o potencial de Redes Neurais Convolucionais (RNC) como ferramenta automatizada para detecção de trincas em superfícies de pavimentos. Foram utilizadas fotografias da superfície de diferentes segmentos de um pavimento do tipo *Chipseal*, obtidas a partir de câmeras fotográficas montadas em veículos. As imagens foram avaliadas a partir da proposta do uso de duas arquiteturas de redes neurais convolucionais e implementadas com o auxílio da biblioteca de aprendizado de máquina PyTorch, o qual possui código aberto e disponível na forma de *script* em linguagem Python. As imagens foram processadas com o uso de três técnicas diferentes, com o intuito de avaliar a influência da complexidade dos algoritmos propostos. Para análise da performance da rede neural, foram utilizadas como métricas de avaliação a acurácia, a precisão, o *recall* e o *F1 score*. Os resultados apontaram que a arquitetura da rede neural escolhida apresentou desempenho satisfatório na detecção de trincas, bem como indicam que a complexidade da rede é um dos fatores a ser considerado durante o processo de classificação das imagens.

1. INTRODUCTION AND BACKGROUND

The pavement distress evaluation is a crucial stage of implementing Pavement Management System (PMS) at the network analysis level. Despite needing fast and accurate data acquisition during the field inspections, the distress evaluation in Brazil still involves manual procedures, which are carried out by trained professionals over randomly selected sections. The collected

and classified data allow the pavement managers to objectively evaluate pavement surface condition. However, this procedure usually demands a considerable time of inspection and is susceptible to human error, which represents considerable costs and does not cover all the network extension (Li *et al.*, 2017; Sun *et al.*, 2009).

The recent advances in sensors, such as photographic cameras and LiDAR (Light Detection And Ranging) devices, combined with advances in data storage cloud computing and data storage, have allowed performing surface inspection of the entire pavement network in an automated manner. Additional benefits of automated pavement distress assessment include minimal traffic impact, reduction of the inspector's safety risks during the field operations, as well as faster evaluation process (Pierce and Weitzel, 2019). These benefits have encouraged transportation agencies around the world to choose this automated process.

Although the semi-automated inspection systems can carry out a continuously high-quality photographic record over all the roadway network, the raw dataset is normally evaluated on the computer screen by handwork. And, considering the schedule, technical and economic issues required in the PMS, the manual distress evaluation of such large amount of data is not feasible.

The pavement surface evaluation by automated procedures has been a challenge for public and private segments in Brazil and many other countries. Even though recent advances on hardware devices, such as laser, optical sensors and high-end computers, can provide basic requirements established in different protocols (e.g. NCHRP-531), the development of high-level intelligent algorithms endures the highest challenge observed in the literature (Fan *et al.*, 2018; Zhang *et al.*, 2017; Zhang *et al.*, 2016).

Recently, studies about efficient high-level algorithms to detect the pavement surface distress have been the focus of different research (Fan *et al.*, 2018; Koch *et al.*, 2015; Ong *et al.*, 2011; Zhang *et al.*, 2017; Zhang *et al.*, 2016). Most of these studies have developed algorithms applied to identify potential cracks in images of pavement surface, which can recognize and quantify the variation of greyscale tones, using different filters, template matching, edge detection, and other image features.

The Machine Learning algorithms applied to automate the inspection of different infrastructures (pavements, bridges, dams, etc.) have become an appropriate option to improve the data analysis speed. This technique consists of data classification using deep learning algorithms, such as the Artificial Neural Network (ANN). The ANN is a group of procedures that reproduce the way that a human brain processes any kind of information, which is composed of different connections, defined as neurons or nodes (Haykin, 2009). For image classification, the Convolutional Neural Networks (CNN), a type of ANN, is usually chosen.

The CNN architecture is analog to human brain neurons connection and inspired in the visual cortex framework. In this case, each neuron responds to stimuli only in restricted areas of field vision, which is defined as the receptive field. The observed image in this space is considered mathematically as a pixel matrix. Every single matrix element can be divided into three color channels labeled as RGB (Red – R; Green – G; Blue – B). The convolution process as observed in Figure 1, allows for extracting exclusive features from the images through the cross-correlation between the filtered (output) and the original (input) images, as well as a specific filter (kernel). Cross-correlation involves a group of matrix operations (Khan *et al.*, 2018), wherein after performed several times can reduce the original matrix into a single vector or dense layer. Due to the matrix multiplication process between the input image and the filter, it is assumed that both matrices have the same element size. The elements of the new matrix (output) are summed up

to represent a unique result that describes a specific feature. This operation is repeated several times by the filter moves (stride operation) over the input matrix elements. Also, this process is useful for obtaining the compressed representation of image features, which is invariant to moderate object changes such as scale, pose (a combination of object orientation and position), and translation over the image.

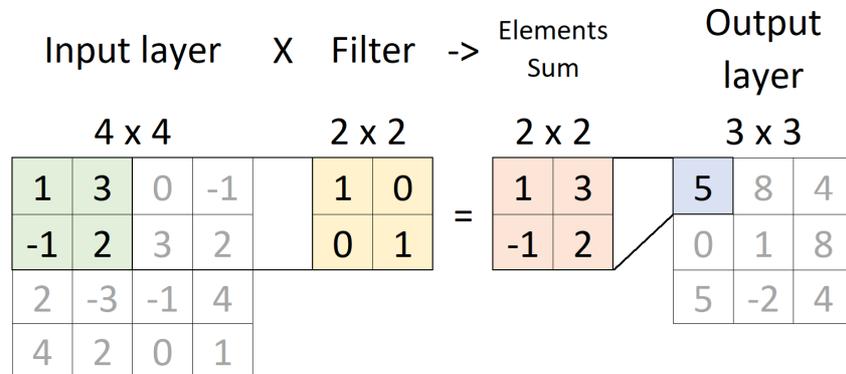


Figure 1. An example of filter application during a convolutional calculation.

The application of neural networks in different infrastructure condition evaluations, including pavement distress surveys, has become more common recently. Previous studies (Fan et al., 2018; Koch et al., 2015; Ong et al., 2011; Zhang et al., 2017; Zhang et al., 2016) have discussed the great potential of this technique in identifying different types of cracks on asphalt or concrete pavements. Moreover, the latest data storage capacity available in the cloud or personal computers associated with parallel computing has permitted acquiring and analyzing large datasets of the roadway network.

This research discusses the performance of two CNN architectures, applied to the automated recognition of cracks in asphalt pavements, from a group of high-resolution images and based on statistical metrics. Besides, the effects of the application of different filters are analyzed, as well as the advantages and limitations of the proposed tool.

2. METHODOLOGY

2.1. 2D image dataset

2D photographic records of the surface of chip seal (a common asphalt surface treatment technique) were used in this study. These images were taken from different segments in the RELLIS Campus, which is part of Texas A&M University in the USA. The downward-facing cameras mounted on vehicles were used for pavement surface images acquisition. Recently, it is common to have other sensor types mounted on these vehicles, such as GNSS (Global Navigation Satellite System) and inertial accelerometer for geolocation history, and digital laser profiler for both pavement cross-section and roughness. The minimum requirements for pavement surface evaluation using semi or fully automated systems are illustrated in the literature (e.g. Pierce and Weitzel, 2019).

Six high-resolution images with 2048 x 8192 pixels in RGB (Red, Green, Blue) format from different pavement sections were analyzed in this study. Each pavement section has 16.1 meters of length by 4.0 meters of width, and an area equivalent to 64.4 square meters (a total of 386.4 square meters of pavement). Based on the pavement section size, each pixel represents a 1.96 x 1.96 millimeters of pavement surface dimension. These parameters are consistent with the

standards established by different transportation departments in the United States (Dalla Rosa *et al.*, 2016; Pierce and Weitzel, 2019). For this study, each high-resolution image was split into 256 new images with 256 x 256 pixels level resolution, wherein each new image covers a 0.5 x 0.5 meters pavement area. The splitting procedure resulted in 1536 new small images, and expanded the number of images used as input in the training, validation, and performance test of CNN.

The input 256 x 256 pixels images size used in this study is similar to other CNN architectures proposed in the literature, such as VGG (Visual Geometry Group), GoogleNet Inception, Res-Net50, which have applied a 224 x 224 pixels input images. Besides, the rate between the original and split images allowed each section to be fully evaluated. Although some images do not indicate visible cracks, all the 1536 input images were used in the CNN calibration, validation and testing as discussed in item 2.2. Figure 2 shows an example of the pavement surface recorded from a section with a typical crack pattern, while Figure 3 represents the mosaic of small images obtained from the split procedure of the original image, which is discussed in item 2.2 as well.

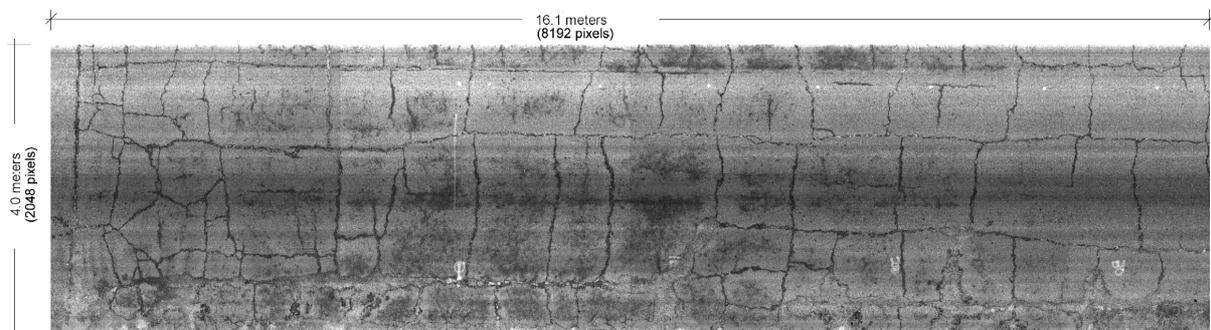


Figure 2. A full-scale image sample measuring 16.1 x 4.0 meters (width x height)

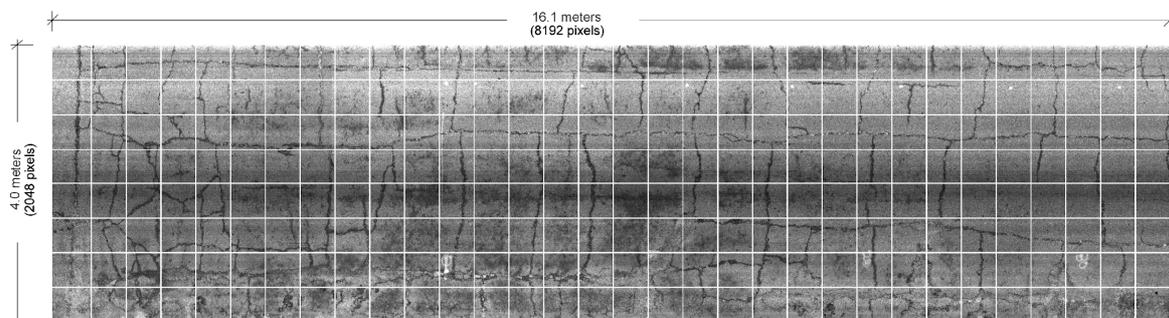


Figure 3. An image mosaic (Figure 2), with 256 x 256 pixels sized or equivalent to 0.5 x 0.5 meters and represented in Figure 5

2.2. Convolutional neural network (CNN) description

A neural network has its structure defined by input and output layers, in addition to the intermediate hidden layers. Each layer is made of neurons (nodes) that process the data. The neurons in each consecutive layer are connected by weights. To transform a traditional feed-forward neural network into a convolutional neural network, which is used in image related works, a feature extractor formed by convolutional layers is added. These layers are represented as volumes. This way, the architecture of a CNN is given by convolutional layers followed by dense layers (fully-connected layers) for the classification task, just like a simple feed-forward neural network

A colored image, with RGB color channels, can be defined as a volume with dimensions given by its height x width x number of color channels. As an example, an image with resolution 256 x 256 pixels and RGB color channels has a volume of 256 x 256 x 3, while a monochromatic image of the same resolution has volume 256 x 256 x 1.

A convolutional layer operates through filters, also called kernels (Figure 1). Each filter must have the same number of channels that it receives as input. It also has a receptive field of a predetermined size, in this case, 3 x 3 (width x height), as shown in Figure 4. It is important to say that each pixel corresponds to 1.96 mm x 1.96 mm and the input image size proposed in the present work corresponds to an approximate area of 0.25 m² (0.5 m x 0.5 m). The CNN architecture proposed in the present work is simpler compared to other architectures as exemplified by (Khan *et al.*, 2018), due to the more simplistic features observed on pavement surfaces.

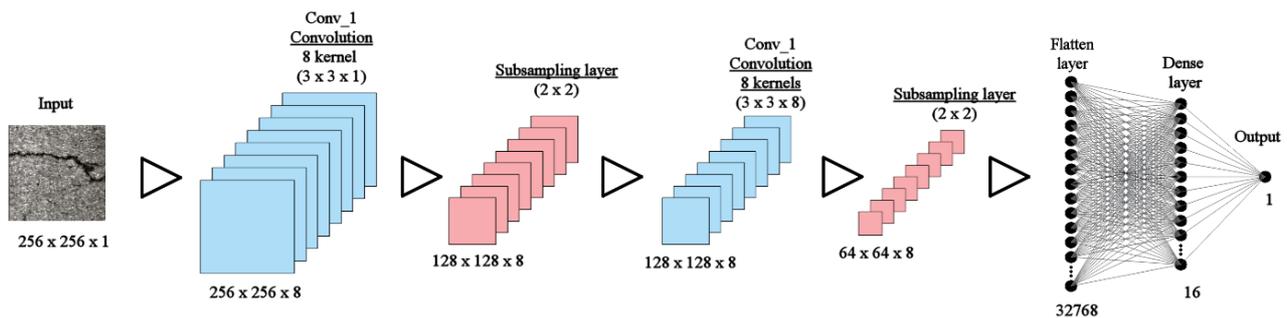


Figure 4. Convolutional Neural Network Used in this study.

Each one of the six images with resolution 2048 x 8192 pixels was divided into 256 images with resolution 256 x 256 pixels as demonstrated in Figure 5. The new group of 1536 images with resolution 256 x 256 pixels was further divided into three subgroups for training, validation, and testing phases. Thus, 1076 (70%) were used during the training process, 230 (15%) were used for validation, and 230 (15%) were used to test the CNN. This split is similar to the ones presented on different previous works that used neural networks (Pianucci *et al.*, 2019; Silva e Lucena, 2018), where the image set was split into 80% for training and validation, and 20% for testing. Also, because the pavement surface has a very dark color, all RGB images were converted to an 8-bit scale, resulting in 256 [0 - 255] possible grayscale values, where the bit 0 corresponds to the color black and 255 corresponds to the color white.

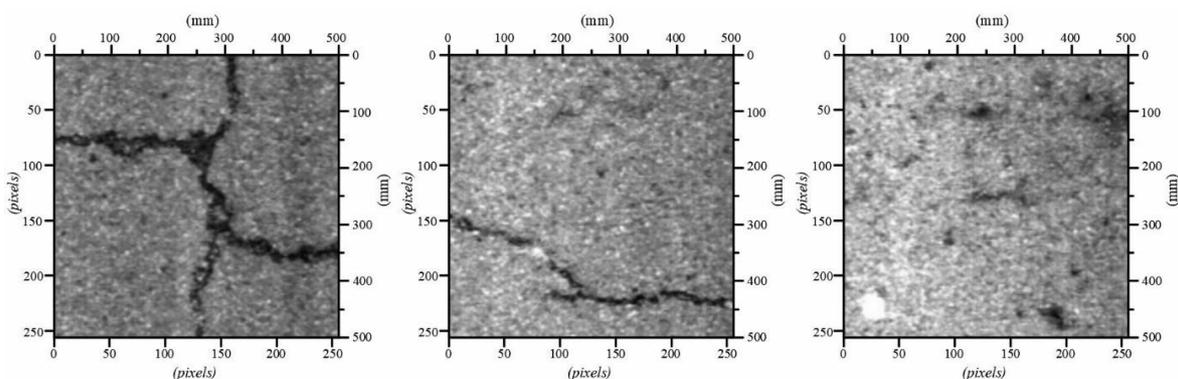


Figure 5: The input dataset samples obtained from the original image and after the split process (Figure 3).

The CNN built for the present work was implemented using the PyTorch library (Paszke *et al.*, 2019), open-source framework for Deep Learning that allows model training with the use of

GPU (Graphics Processing Unit), which makes the training faster due to the parallel execution of processes. PyTorch was released in early 2017 by the Facebook AI Research lab and is already used by other companies. In this work, PyTorch (torch version 1.0.1 and torchvision version 0.2.2) was used to build, train, and evaluate the classification model, utilizing the version 3.6.1 of Python programming language. All the analyses with the CNN were done on a computer with AMD FX(TM)-6300 Six-Core processor, 12 GB of RAM, and one NVIDIA® GeForce GTX 1050 Ti GPU with 4 GB of memory along with the version 9.2 of the CUDA® library.

2.3. CNN architectures and training

Before the training phase, three pre-processing procedures were applied to the analyzed images. In Model A, the Data Augmentations process was used, which aims to virtually increase the number of distinct images to feed the model. This virtual increase in the number of images decreases the chances that the model memorizes the training data and, therefore, increases its generalization capacity. During feeding an image to the model, it can be flipped horizontally and vertically with a 50% probability. As a result, at each iteration, the model will receive as input a slightly modified version of the same image.

Besides the Data Augmentation step (Figure 6a), two filters known as binarization (Figure 6b) and dilation (Figure 6c) were applied to Model B. In the binarization process, the pixel values less than or equal to 40 were set to 255 (white) and pixel values greater than 40 were set to 0 (black). This step was done with the use of the OpenCV library (Bradski, 2000), that is available for Python as well as other languages like C++ and MATLAB.

The learning CNN process for both models A and B was carried out through batches of 16 images. After concluding each epoch, the network performed the updating of its parameters (filters, weights, and biases) to reduce the loss during the machine learning. The loss was calculated by the cross entropy H (Eq. 1), also known as “softmax loss” function, which represents the measure of the distance between the probability of output obtained p_i (forecast) and the probability of expected output y_i . This function allows evaluating how far the neural network is from the expected result.

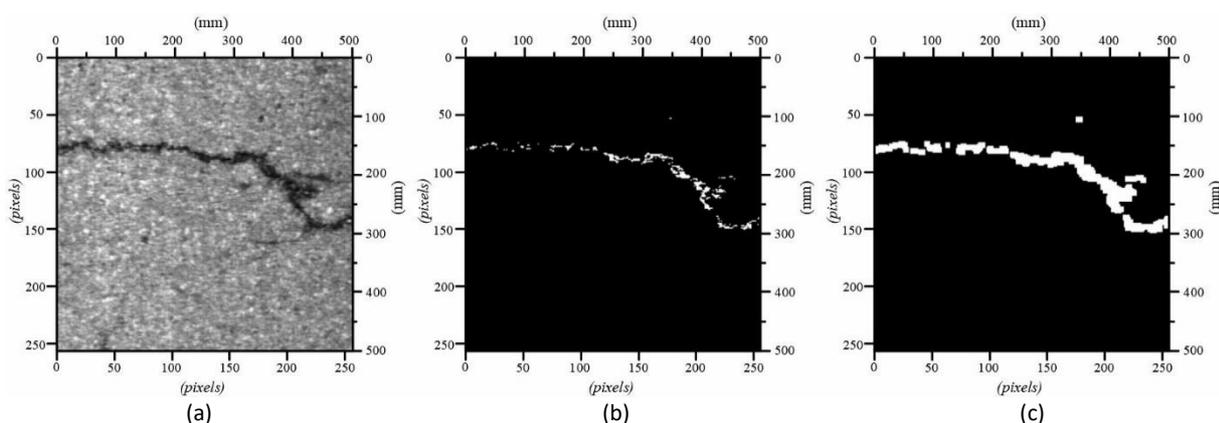


Figure 6: Additional used filters in this study associated with model B: a) Original image; b) Binarization; c) Dilation.

In Eq. 1, the forecast p_i (Eq. 2) is determined by a sigmoid or logistic activation function and classified between 0 and 1, which is connected to the last layer of the network. This logistic function is commonly applied to binary classification in CNN architecture, wherein the x_i represents the output response of the last network layer of i image, while w^T is the weight assigned

to this specific last layer. The weight is calibrated according to the batch of input images inserted into the network. This calibration process is commonly performed by descending gradient algorithms as exemplified in the literature (e.g. Khan *et al.*, 2018).

$$H(p, y) = -\frac{1}{n} \sum_{i=0}^n y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i) \quad n \in [1, N] \quad (1)$$

$$p_i = \frac{1}{1 + e^{-w^T \cdot x_i}} \quad (2)$$

2.4. Evaluation metrics

The quality of the semi or fully automated image acquisition system can be verified by a classification statistic based on a binary confusion matrix (True/False, Positive/Negative, 1/0), with size 2 x 2, normally applied on healthcare diagnosis. This 2 x 2 matrix is made of the number of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), given by the classifications made by the CNN. These binary quantitative results can be used as an additional evaluation method for the quality of a classifier and image acquisition (Dalla Rosa *et al.*, 2016). Besides, previous works (Dung e Anh, 2019; Osman *et al.*, 2019) have proposed the use of these metrics to measure the accuracy (Eq. 3), precision (Eq. 4), recall (Eq. 5) and *F1-Score* (Eq. 6) of a classification model.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F_1 \text{ score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

The accuracy takes into consideration the number of correct predictions in proportion to the number of total examples, indicating how often the classifier is correct in its prediction. However, the analysis solely of this parameter does not convey detailed information considering the case of a CNN applied to image classification. Precision represents the number of correct predictions in proportion to the total number of predictions made by CNN for that class. This metric is often used in a situation where an FP presents more harm than FN classification. The recall is given by the frequency at which the classifier detects examples of a given class. In the context of this work, recall is then defined as the ratio between the number of correct predictions of a crack in a pavement surface and the total number of examples containing a crack. A recall analysis is useful to determine how big is the number of False Negatives, therefore widely used in situations where an FN can cause more harm in a classification task. The *F1-Score* is defined as the harmonic average between precision and recall, measuring the performance of a model. High values of *F1-Score* indicate low FP and FN counts, meaning the classifier is doing a good job at its task.

3. RESULTS AND DISCUSSION

3.1. Training and validation performance test

To evaluate the CNN performance during training, validation, and test images sets, one hundred epochs were set up. The epoch number represents how many times an image batch passes through the neural network. As observed in Fig. 7a, there is a clear trend to Model A to achieve a lower cross-entropy loss during training faster than the validation procedure. This evidence indicates that the initial model architecture “memorizes” the input dataset during training phase, which has resulted in the CNN failure to generalize the used images in the validation analysis. This behavior was observed for both proposed models in this research.

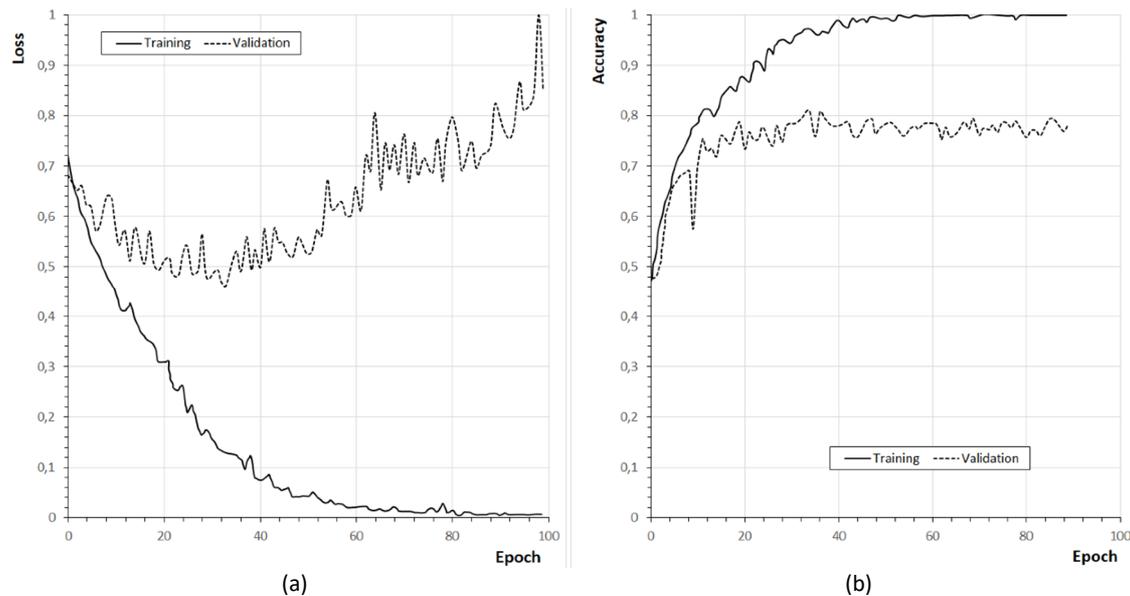


Figure 7. The Model A performance evaluation: a) cross-entropy loss; b) accuracy.

Another finding that supports the conclusion that image memorization by CNN has occurred is related to the maximum accuracy, as observed in Fig. 7b. The highest detected accuracy during the validation procedure was 20% less than identified during training, which emphasizes this issue. The neural network memorization is discussed in the literature (e.g. Arpit et al., 2017; Haykin, 2009). Certain specific characteristics (even noise) present in the dataset may reduce the network’s ability to generalize the relationship between input and output and result in a data over adjustment or overfitting. The over adjustment can be defined by the error decrease during the training, while the test group error remains elevated.

Furthermore, the generalization issue could be influenced by the size and how representative is the training dataset, as well as the neural network architecture complexity. The complexity is defined in this study as the number of layers that the CNN was configured. Based on this evidence, the initially proposed architecture was reviewed to make it compatible with the number of CNN layers and the size of the dataset.

To avoid possible memorization during the neural network training, an additional image transformation was added in both models in the Data Augmentation step, which is named in this study as “Modified Architecture” and represented in Figure 4. To perform this new step, a random image rotation action was introduced before the mirroring procedure in the CNN architecture. The image random rotation adopted in this study ranges from -20 to +20 degrees as exemplified in Figure 8.

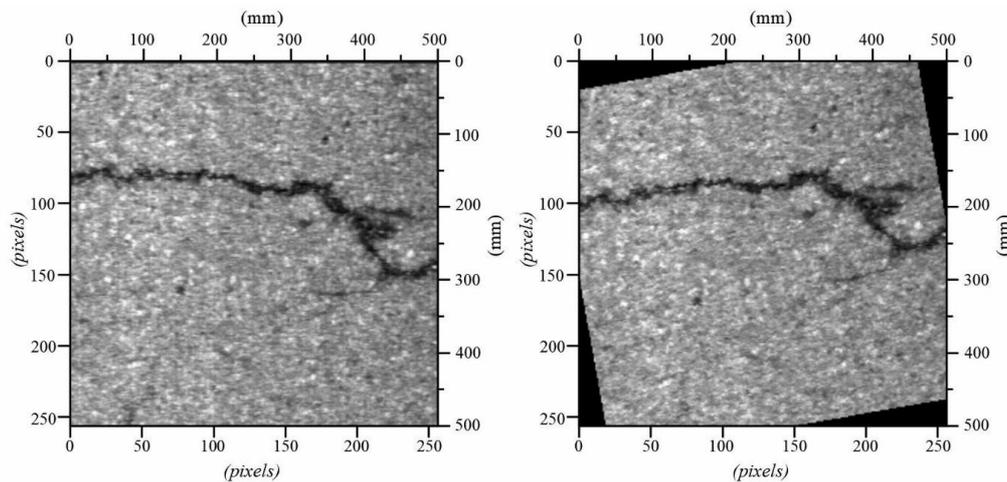


Figure 8. An image sample after the random rotation due to the new Data Augmentation procedure

Also, the number of filters in the Modified Architecture was reduced from 32 and 16, respectively for the first and second convolutional layers, to 8 eight filters each one. Additionally, the number of connection nodes between the dense layer with the layer immediately before the output layer was resized from 256 to 16. These improvements allowed to decrease the model's complexity and the computational memory demand, which was reduced from 393 to 6.1 Mega-bytes (about 98% lower) during the training. Also, the necessary inference time for each image during the validation analysis dropped from 104 to 30 milliseconds.

Additionally, to prevent the overfitting issue during the training procedure, the dropout was increased by 50% in the dense layers. The dropout indicates that some neurons (nodes) of the neural network can be ignored during the training, which reduces the CNN ability to learn the problem. To anticipate slower learning due to reduced model complexity, it was necessary to raise the number of epochs from 100 to 2000.

These implemented procedures to reduce both model's complexities have allowed improving the CNN performance compared to the first architecture, due to a more consistent validation loss during the neural network training. Particularly, the new model A architecture achieved a minimum cross-entropy loss in lower validation (45%) compared to the first algorithm (82%). This evidence may indicate that, during the training procedure, there was a good generalization acquired for CNN, as well as the information learned by the neural network from the training dataset has been more efficiently applied in the validation step.

3.2. Evaluation metrics

To evaluate the CNN performance of the trained models A and B architectures, the test dataset was applied. The metrics for both models are summarized in Table 1. The epoch 30 was defined to assess the metrics for Model A (Figure 5) before the algorithm's improvement, which corresponds to the lowest cross-entropy loss during the validation procedure. Also, only the classified images with a probability higher than 70% were considered as cracked pavement. To compare the performance of the developed models with models developed in previous studies, the metrics observed by other studies are also presented in Table 1 (Fan et al., 2018; Osman et al., 2019; Zhang et al., 2016).

The classification results obtained from CNN with the test dataset indicates that Model A achieved an accuracy greater than 76% using the initially proposed architecture. However, after

the suggested improvements, the modified Model A algorithm reaches a performance of nearly 84%, which corresponds to 193 (TP=100; TN=93) images that were correctly classified. Although this behavior was also observed in Model B, a fewer expressive performance gain compared to Model A was identified. This evidence indicates that both image binarization and dilation procedures do not contribute to CNN overall performance.

Table 1 – Metrics evaluation for different the CNN models proposed in this study

Performance	Evaluated models				Previous studies		
	A	A (Modified)	B	B (Modified)	Zhang et al. (2016)	Fan et al. (2018)	Osman et al. (2019)
False Positives (FP)	38	21	33	23	-	-	-
False Negatives (FN)	17	16	16	23	-	-	-
True Positives (TP)	99	100	100	93	-	-	-
True Negatives (TN)	76	93	81	91	-	-	-
Accuracy	76,1%	83,9%	78,7%	80%	-	98,5%	99%
Precision	72,3%	82,6%	75,2%	80,2%	86,9%	91,8%	99,1%
Recall	85,3%	86,2%	86,2%	80,2%	92,5%	88,1%	98,1%
F_1 Score	78,3%	84,4%	80,3%	80,2%	89,6%	89,9%	98,6%

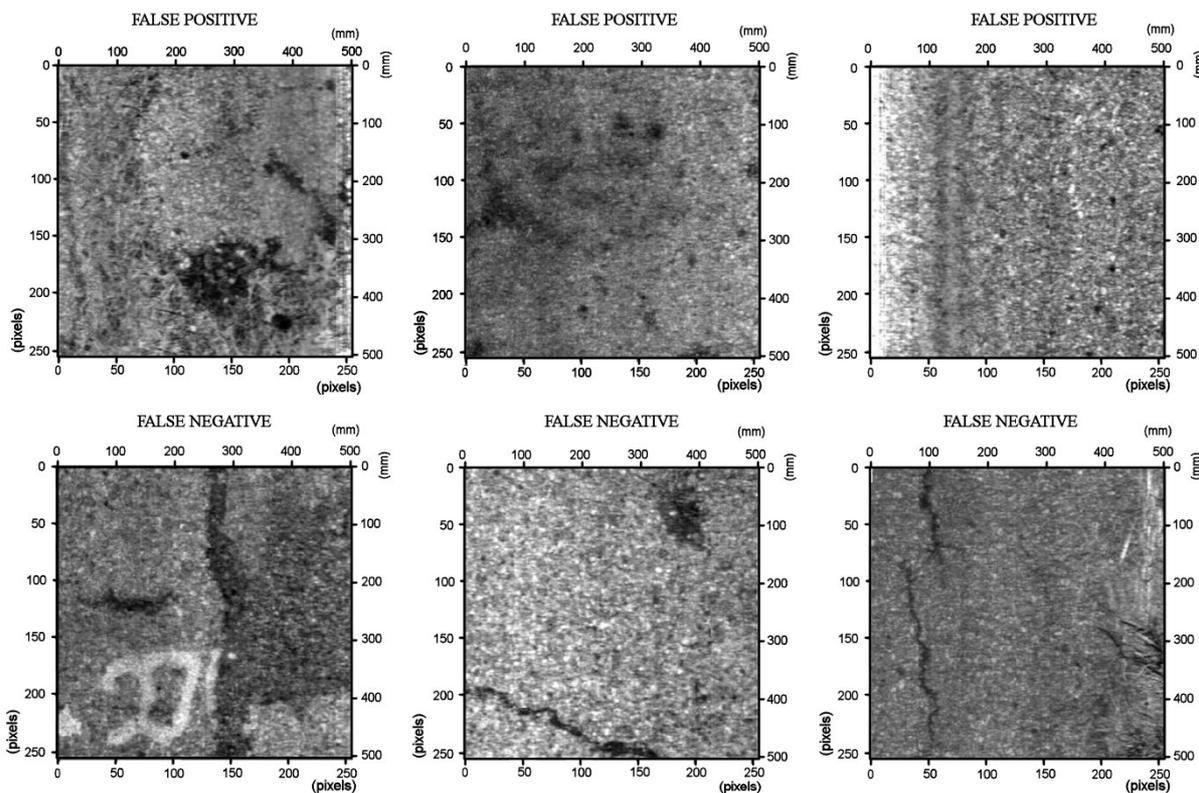


Figure 9: Examples of incorrect classifications observed during the CNN analysis.

Figure 9 shows typical situations of different images that were incorrectly classified by the CNN, resulting in FP or FN. In this study, the higher number of FP than FN for both architectures are noticed in Table 1. The implementation of additional Data Augmentation step in the modified architectures was able to reduce the number of FP by 55% for Model A, and approximately 70% for Model B. Nevertheless, the number of FN remained unchanged for Model A, while Model B showed an increase of images without cracks that were incorrectly classified. This particular situation resulted in relatively lower precision than the recall parameter, which indicates

that both models tend to benefit the Positive class. Also, this outcome shows that the CNN considers the largest number of cracks more important, even though this process results in FN as well.

Another important evidence about the model's overall performance was established by the F_1 score (Table 1). The algorithms adjustments carried out in both models only promote an improvement in Model A, leading to the best result ($F_1=84.4\%$) as well as define the chosen CNN architecture. Also, the best F_1 score observed in this study is relatively close to values obtained by previous studies published in the literature (Fan et al., 2018; Osman et al., 2019; Zhang et al., 2016). While the CNN implemented here using the PyTorch library have shown a great potential for automated asphalt pavement crack detection systems, a more robust group of images available on the training procedure could help to improve the overall performance.

4. CONCLUSION

This study presents the potential of Convolutional Neural Networks (CNN) as an automated tool for crack detection in asphalt pavements. Six high-resolution images were split into 1536 images with 256 x 256 pixels resolution, which were used for training, validating, and testing the developed architectures. The results show that the proposed CNN algorithms have a satisfactory performance in detecting surface cracks in asphalt pavement images. Specific remarks and recommendations are discussed below.

The dimension of input images used in CNN was adequate for the training, validation, and testing steps. Although the images with dimensions lower than 256 x 256 pixels will allow for reducing the necessary processing time, they would not be able to improve the ability of CNN generalization because essential features, such as edges detection, could be missed. On the other hand, larger image sizes can result in high computational cost, which may be unsuitable for application in fully automated pavement defects detection systems.

Even though CNN shows a high ability to learn the image features patterns, which resulted in a good model generalization, possible issues encountered by the network could have affected its overall performance. Usually, surface texture, traffic marking or the presence of liquid substances (spots) on the pavement surface, as well as the weather conditions can directly affect the asphalt pavement distress classification. Also, other defects such as potholes may result in a larger number of FP. This evidence indicates that CNN gives high importance to the edges and color contrast over the pavement surface.

CNN's skill to identify other types of cracks, particularly those caused by fatigue mechanism, is still considered limited due to the image pattern used in this study. Usually, pavements that experience the fatigue mechanism show an irregular and random crack distribution, which is different from the pattern used for CNN training. Feeding the proposed CNN with larger and more diverse images dataset, with distinct types of cracks, will improve the generalization network ability. Future studies could use a larger image dataset and consider different crack types and severity levels. Such future work could improve the network's ability to classify different cracking patterns, including earlier stage cracks where the opening is still relatively small.

Despite the collection of images by automated systems covers a relatively small area, the overlap of each pixel on the pavement surface should be considered the main aspect to be evaluated. The resolution of the images used in this study, which corresponds to a squared area with 1.96 mm x 1.96 mm, was appropriate for the CNN classification procedure. However, public and private transportation agencies need to define the minimum camera requirements to maximize

the efficiency of the algorithms. Therefore, new protocols that establish the minimum requirements for pavement surveys using automated systems are essential.

A high number of filters (layers) may increase the probability that the CNN memorizes the image's features during the training step. The memorization issue can result in a performance loss during the network validation and testing. This evidence is supported by the evaluated metrics, which indicated a clear drop in the number of images that were incorrectly classified by the modified architecture. The adoption of a simplified network may improve the ability to identify the defects observed in different images and a considerable computational cost reduction during the CNN classification.

Currently, code libraries, such as the one used in this study, have allowed CNN to use the parallel processing power, available on modern graphics cards (GPUs), in which the number of processing units is higher at a relatively low cost. According to the manufacture specifications, the entry-level GPU used in this study has 768 processing units (Stream Processors), which is considerably higher than what is available in the current Central Processing Unit (CPU). By paralleling the hardware, the algorithm could be improved to consider both the type and severity levels of pavement defects, a substantial improvement to the input data quality for Pavement Management Systems (PMSs). Also, the qualified automated pavement inspection system will contribute to a more efficient prediction of pavement performance at the network-level. Future work could investigate the impact on the results of the analyses performed within PMSs.

REFERENCES

- Arpit, D.; S. Jastrzębski; N. Ballas; D. Krueger; E. Bengio; M. S. Kanwal; T. Maharaj; A. Fischer; A. Courville; Y. Bengio and S. Lacoste-Julien (2017). A Closer Look at Memorization in Deep Networks. Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. *ArXiv - a repository of electronic preprints*, 1–10. arXiv:1706.05394
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. Available at: <http://drdobbs.com/open-source/184404319>
- Dalla Rosa, F.; N. G. Gharabeh; E. G. Fernando and A. Wimsatt (2016). Quality Assurance for Automated and Semi-Automated Pavement Condition Surveys. *International Conference on Transportation and Development 2016*. p. 192–201. doi:10.1061/9780784479926.018
- Dung, C. V. and L. D. Anh (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, v.99, p. 52–58. doi:10.1016/j.autcon.2018.11.028
- Fan, Z.; S. Member; Y. Wu; J. Lu and W. Li (2018). Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network. *ArXiv - a repository of electronic preprints*, p. 1–9. arXiv:1802.02208
- Haykin, S. (2009). *Neural networks and learning machines*. (3rd ed). Pearson, Ontario.
- Khan, S.; H. Rahmani; S. A. A. Shah and M. Bennamoun (2018). A Guide to Convolutional Neural Networks for Computer Vision. *Synthesis Lectures on Computer Vision*, v.8 n.1, p. 1–207. doi:10.2200/s00822ed1v01y201712cov015
- Koch, C.; K. Georgieva; V. Kasireddy; B. Akinci and P. Fieguth (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, v.29, n.2, p. 196–210. doi:10.1016/j.aei.2015.01.008
- Li, S.; Y. Cao and H. Cai. (2017). Automatic Pavement-Crack Detection and Segmentation Based on Steerable Matched Filtering and an Active Contour Model. *Journal of Computing in Civil Engineering*, v.31, n.5, doi:10.1061/(ASCE)CP.1943-5487.0000695
- Ong, G. P.; S. Noureldin and K. Sinha (2011). Technical report: Automated Pavement Condition Data Collection Quality Control, Quality Assurance, and Reliability. doi:10.5703/1288284314288
- Osman, M. K.; M. H. M. Noor; A. Ibrahim; N. M. Tahir; N. M. Yusof and N. Z. Abidin (2019). Deep convolution neural network for crack detection on asphalt pavement. *International Conference on Nanomaterials: Science, Engineering and Technology (ICoNSET) 2019*. v. 1349, doi:10.1088/1742-6596/1349/1/012020
- Paszke, A.; S. Gross; F. Massa; A. Lerer; J. Bradbury; G. Chanan; T. Killeen; Z. Lin; N. Gimelshein; L. Antiga; A. Desmaison; A. Kopf; E. Yang; Z. DeVito; M. Raison; A. Tejani; S. Chilamkurthy; B. Steiner; L. Fang; J. Bai and S. Chintala (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, & R. Garnett (Eds), *Advances in Neural Information Processing Systems 32* (p. 8024–8035). Available at: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>

- Pianucci, M. N.; C. S. Pitombo and A. L. Cunha (2019) Previsão da demanda por viagens domiciliares através de método sequencial baseado em população sintética e redes neurais artificiais. v. 27, n.4, p. 1–23. doi:10.14295/transportes.v27i4.1406
- Pierce, L. M. and N. D. Weitzel (2019). Automated Pavement Condition Surveys. Automated Pavement Condition Surveys. Transportation Research Board, Washington, D.C. doi:10.17226/25513
- Silva, W. R. L. and D. S. Lucena (2018). Concrete Cracks Detection Based on Deep Learning Image Classification. Proceedings, v.2, n.8. doi:10.3390/icem18-05387
- Sun, Y.; E. Salari, and E. Chou (2009). Automated pavement distress detection using advanced image processing techniques. Proceedings of 2009 IEEE International Conference on Electro/Information Technology. EIT 2009. p. 373–377. doi:10.1109/EIT.2009.5189645
- Zhang, A.; K. C. P. Wang; B. Li; E. Yang; X. Dai; Y. Peng; Y. Fei; Y. Liu; J. Q. Li and C. Chen (2017). Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network. Computer-Aided Civil and Infrastructure Engineering, v.32, n.10, p. 805–819. doi:10.1111/mice.12297
- Zhang, L.; F. Yang; Y. D. Zhang and Y. J. Zhu (2016). Road crack detection using deep convolutional neural network. Proceedings - International Conference on Image Processing (ICIP). p. 3708–3712. doi:10.1109/ICIP.2016.7533052